



NRC Publications Archive Archives des publications du CNRC

A Knowledge Representation Model For Matchmaking Systems in e-Marketplaces

Joshi, Manish; Bhavsar, Virendra; Boley, Harold

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. /
La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

Publisher's version / Version de l'éditeur:

*Proceedings of the 11th International Conference on Electronic Commerce ICEC
2009, 2009-08-15*

NRC Publications Record / Notice d'Archives des publications de CNRC:

<https://nrc-publications.canada.ca/eng/view/object/?id=d6c68ad0-f578-4dfd-a45d-9c8b6a4c627a>
<https://publications-cnrc.canada.ca/fra/voir/objet/?id=d6c68ad0-f578-4dfd-a45d-9c8b6a4c627a>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at

<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site

<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at

PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.



A KNOWLEDGE REPRESENTATION MODEL FOR MATCHMAKING SYSTEMS IN e-MARKETPLACES

Manish Joshi

University of New Brunswick
Faculty of Computer Science
Fredericton, Canada
joshmanish AT gmail.com

Virendra Bhavsar

University of New Brunswick
Faculty of Computer Science
Fredericton, Canada
bhavsar AT unb.ca

Harold Boley

National Research Council
Institute for Information Technology
Fredericton, Canada
Harold.Boley AT nrc.gc.ca

ABSTRACT

The success of matchmaking systems largely depends on how effectively product/service descriptions (profiles) of participants are modelled. We formalize the multifaceted expectations and interests of participants as ‘constraints’ in those profiles. We identify and explicitly define the relevant types of constraints. We propose a new knowledge representation (KR) model for Web-based matchmaking systems that can represent these constraints. We present a system that implements the proposed KR model, exemplifying its features and evaluating its performance.

Categories and Subject Descriptors

I.2.4 [Knowledge Representation Formalisms and Methods]

H.3.5 [On-line Information Services]

General Terms

Algorithms, Performance, Design, Economics, Experimentation

Keywords

Knowledge Representation Model, Matchmaking in e-marketplaces, Multifaceted Constraints

1. INTRODUCTION

Matchmaking is considered here as the process of optimally pairing up participants from two groups, typically called *sellers* and *buyers*, according to an optimality criterion formalized as a similarity value.

In e-marketplaces, all participating sellers and buyers submit their profiles (containing descriptions of products/services offered and sought, including preferences) and wish to get a ranked list of matching profiles of other participants, typically from the other group.

Participants’ expectations, formalized as *constraints*, can be quite numerous and multifaceted, which make them difficult to model.

In section 2, we discuss several aspects of matchmaking in e-marketplaces that arise due to the complex nature of participants’ expectations. The performance of a matchmaking system largely depends on how well it supports the various types of constraints. Section 3 discusses various KR models used to develop

matchmaking systems. None of these models represents constraints of all the types discussed in section 2. We propose a new KR model that captures multifaceted constraints in participant profiles, leading to the development of an effective matchmaking system. Details of the proposed KR model are given in section 4. Section 5 presents a matchmaking algorithm for the proposed KR model. Key features of the matchmaking system based on the proposed KR model are discussed in section 6. An evaluation is given in section 7, and our conclusion, in section 8.

2. ASPECTS OF MATCHMAKING

Based on [5], we discuss the multifaceted nature of constraints in section 2.1 and characterize additional aspects of matchmaking in section 2.2.

2.1 Multifaceted Constraints

A constraint is a condition on a profile facet (‘feature’, ‘attribute’). In the literature, mostly *hard* and *soft* constraints have been defined explicitly [9, 13]. In this section we define and describe additional types of constraints. In subsequent sections we elaborate how our proposed model represents all these types of constraints in combination.

a) Hard and Soft Constraints: These terms reflect the relative flexibility of participants regarding the fulfilment of a constraint. In case of a soft constraint, a participant is ready to proceed with a match even if the facet value described by his/her constraint is not satisfied by the facet value of the corresponding constraint of the counterpart profile. In contrast, a participant does not compromise with an offer/request specified as a hard constraint.

b) Range Value Constraints: Participants involved in matchmaking often provide a range for their offerings rather than a discrete value, e.g. ‘Looking for an apartment with the rent 500 to 600’. This constraint should be matched with all constraints of other participants that offer the rent in the range of 500 to 600.

c) Multi-Valued Constraints: Participants sometimes specify multiple discrete values as their (disjunctive) choices. For example, a constraint ‘I want a shared or a single apartment’ should be matched with all constraints offering a shared apartment as well as with all constraints offering a single apartment.

d) Preferential Constraints: For the soft constraints of a profile, a participant may wish to indicate relative preferences among various facets. For example, consider a participant’s apartment profile with rent facet preferred to facets area, type, pet-allowed. This profile can succeed in spite of low constraint satisfactions for the other facets as long as the rent constraint is highly satisfied. Bassiliades et al. [10] have discussed the preferences of facets in the context of apartment renting domain.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICEC ’09, August 12-15, 2009, Taipei, Taiwan

Copyright © 2009 ACM 978-1-60558-586-4/09/08...\$10.00.

e) *Hidden Cost Constraints*: In e-business matchmaking, cost is an important facet that affects successful outcomes. Some participants (especially from the seller group) may hide facet values that could increase the cost. For example, a constraint formalizing “the rent of the apartment is \$550, electricity extra”, should not succeed with the constraint of a participant who seeks a rent of \$550

2.2 Matchmaking results

Based on characteristics of the matchmaking systems’ results, a few more aspects of matchmaking are identified.

a) *Compromise match effect*: A concept of soft constraints leads to the notion of a compromise match. Two constraints from two profiles have a compromise match if

- i) either one or both of the constraints in comparison are soft constraints, and
- ii) the values of the facets of both the corresponding constraints do not match.

Different matchmaking systems have different strategies to resolve the issue of compromise matches.

b) *Symmetric / Non-symmetric*: If a matchmaking system returns identical results of matching a profile P_1 with P_2 and matching a profile P_2 with P_1 , then the system is called a symmetric system, otherwise it is a non-symmetric system.

For example, let the profile P_1 have a `security-deposit` facet and the profile P_2 be without such a facet. A symmetric matchmaking system results in identical similarity values when P_1 is compared with P_2 and when P_2 is compared with P_1 . In contrast, a non-symmetric matchmaking system results in different similarity values as a consequence of these comparisons.

c) *Result Classification Categories*: A participant may not be interested to have a list of all matching profiles as the result of a matchmaking process, especially when the numbers of profiles in the result is large. A participant wishes a ranked list of matching profiles preferably grouped in specific categories.

3. VARIOUS KR MODELS

Matchmaking systems use some KR model to represent participants’ profiles. We discuss various KR models and matchmaking systems developed using these models, in following subsections.

3.1 Array (Vector) of Features

This is a basic KR model used in early matchmaking systems. Participants’ profiles are stored either in the form of documents, a database or in a file using XML. Keywords extracted from documents are used for matchmaking among documents. A typical Information Retrieval (IR) methodology is used as the basis of matchmaking. The COINS [4] and the GRAPPA [13] matchmaking systems use this KR model.

3.2 Knowledge Representation Languages

KR languages are used to represent the concept definitions of an application domain in a structured and formally well-understood way [1]. Matchmaking systems based on KR languages emphasize the semantics, in contrast to earlier matchmaking systems which focused on the frequency of keywords. Several matchmaking

systems use description logic to model domain knowledge. A semantic reasoner is used for matchmaking in some systems while other systems use customized algorithms for matchmaking. The LARKS based system [12], the Description Logic based NeoClassic Reasoner [8] and the Semantic Web language DAML-S based system [6] use KR languages to represent knowledge.

3.3 Tree

Some researchers have proposed the use of a tree structure to represent knowledge. Islam et al. [3] used a basic tree structure and proposed a matchmaking framework to identify a set of matching resources for a job, from a large collection of resources in a distributed environment. Bhavsar et al. [2] developed a matchmaking system that uses node labelled, arc labelled, arc weighted trees to represent knowledge.

3.4 Graph

Like in a tree structure, nodes and edges of a graph are used to represent concepts and relationship among these concepts. Mohaghegh et al. [7] proposed a matchmaking system in the domain of online recruitment. The IMPACT system [11] uses graph to represent knowledge.

3.5 Hybrid

A combination of different techniques is used to represent participants’ information. Ragone et al. [9] proposed a semantic matchmaking approach that integrates various knowledge representation technologies. It uses a combination of DLR-Lite, fuzzy rules, and utility theory to represent profiles of participants.

4. PROPOSED MODEL

We propose to represent a participant profile as a set of constraints, such that $P = \{C_1, C_2, C_3, \dots, C_m\}$. Each constraint is a quadruple $C_i = \langle a, d, f, p \rangle$, where a is an attribute, d is a set of values to describe an attribute, f indicates the flexibleness of a constraint and p is the priority of a constraint. All elements of a constraint are described below.

Attribute (a) – An *attribute* represents the facet. For example, if a participant has a constraint ‘need 4 bedrooms’, then the attribute of this constraint is ‘bedrooms’. This field always has an alphabetical value.

Description (d) – The *description* represents a set of values assigned to an attribute of a constraint. In a constraint ‘need 4 bedrooms’, the attribute ‘bedrooms’ has the description value ‘4’. Let D be the domain of d . $d \subset D$. D contains alphabetical strings or numerical values or a combination of both or a range value having format like $num1 \dots num2$ such that $num1, num2 \in R$.

As the description is a set of values, it can represent *multi-valued constraints*. A set of constraints ‘looking for a shared or a bachelor apartment’, ‘rent is 1500’, ‘available from September-1’, and ‘pets should be allowed’ can be represented as $\langle \text{type}, \{\text{sharedApartment}, \text{BachelorApartment}\}, f, p \rangle$; $\langle \text{rent}, \{1500\}, f, p \rangle$; $\langle \text{availableDate}, \{\text{Sept-01}\}, f, p \rangle$; and $\langle \text{pets}, \{\text{allowed}\}, f, p \rangle$ respectively. In these examples, we have not specified any values of f and p for the constraints.

Consider that a participant asks for ‘2 or 3 bedroom apartment’. In this case the attribute ‘bedrooms’ has the description value that can be represented as a set of ‘multiple values’ or a ‘range’. Hence $\langle \text{bedrooms}, \{2, 3\}, \text{Yes}, p \rangle$ and $\langle \text{bedrooms}, \{2 \dots 3\}, \text{Yes}, p \rangle$ are both valid representations and have identical meanings. Figure 1 shows a rent constraint that has the description as the range.

Flexibility (f) – The *flexibility* indicates whether a constraint is a hard or a soft constraint. $f \in F$, where $F = \{\text{No}, \text{Yes}\}$. A ‘No’ value of f (i.e. no flexibility) indicates a hard constraint, whereas a value ‘Yes’ represents a soft constraint. The constraint of a buyer ‘house rent must be 500’ indicates a hard constraint and is represented as $\langle \text{rent}, \{500\}, \text{No}, p \rangle$. A constraint, ‘Smoking is not allowed, but can smoke in balcony’ represents a soft constraint. It can be represented as $\langle \text{Smoking}, \{\text{Not allowed}\}, \text{Yes}, p \rangle$.

Priority (p) – The *priority* describes the relative priority of soft constraints among other soft constraints, in a profile. The value of p can be any real value greater than 0. $p \in R$. All soft constraints are initialized with the priority values of 1. The priority values for all soft constraints are set automatically to match the preferences indicated by participants.

For example, if a buyer specifies that pets allowed facet is more important to him than all remaining facets, then priority value for this constraint is set to a value greater than 1. The constraint is represented as $\langle \text{pets}, \{\text{allowed}\}, \text{No}, 1.1 \rangle$, and all remaining constraints will have p values as 1. These priority values ultimately used to rank the service represented by the facet.

Figure 1 illustrates how a tenant’s (buyer’s) profile is represented using our KR model. The description provided by the tenant is followed by a quadruple representation of constraints.

Profile – Tenant (Buyer)

I am a mature student looking for an affordable shared or single apartment on the south side of Fredericton for September. Finishing up my last year at UNB, I smoke but can adjust with non-smoking apartment. rent – 400 to 450. Please contact if anything is available, thanks!

```
<type, {apartment, shared}, No, 1>
<rent, {400...450}, Yes, 1>
<area, {South side}, No, 1>
<smoke, {allowed}, Yes, 1>
<available, {Sept-01}, No, 1>
```

Figure 1. Representation of the constraints of a buyer

Next section describes an algorithm for calculating similarity between any two profiles.

5. ALGORITHM

The similarity value between any two profiles is defined as a function of attribute, description, flexibility and priority values of all constraints from both profiles. For any two profiles P_x and P_y , where P_x has m constraints and P_y has n constraints, a similarity value is given by,

$$\text{Sim}(P_x, P_y) = \prod_{i=1 \text{ to } m, j=1 \text{ to } n} S(C_i, C_j) \quad (1)$$

where the function $S(C_i, C_j)$ calculates an intermediate similarity value using steps given in the algorithm below.

The attribute, description, flexibility and priority values of a constraint, are accessed using a notation $C_i.a$ which means the attribute value of the constraint i .

Algorithm

```
1: if (Ci.a = Cj.a) then
2:   if (Ci.d = Cj.d) then
3:     return S(Ci, Cj) = Ci.p × Cj.p
4:   else
5:     if (Ci.f = No) AND (Cj.f = No) then
6:       return S(Ci, Cj) = Ci.p × Cj.p ×
         relativeDifference(Ci.d, Cj.d)
7:     elseif (Ci.f = Yes) AND (Cj.f = Yes)
8:       return S(Ci, Cj) = Ci.p × Cj.p × β
9:     else
10:      return S(Ci, Cj) = Ci.p × Cj.p × α
11:   move on to next Ci and Cj
12: if (Ci.a < Cj.a) then
13:   return S(Ci, Cj) = Omission Penalty
14:   move on to next Ci
15: if (Ci.a > Cj.a) then
16:   return S(Ci, Cj) = Omission Penalty
17:   move on to next Cj
```

The algorithm compares two constraints of two profiles. If the attributes of both the constraints are same then an intermediate similarity value is calculated by checking the description values. If the description values are not same then an intermediate similarity value is calculated by considering the flexibility of the constraints. When hard constraints in two profiles do not match, instead of reducing a similarity value immediately to zero, we compute relative difference between the two corresponding description values of these attributes. A routine *relativeDifference* computes relative difference which is later used to calculate a similarity value. We make sure that an intermediate similarity value for such constraints is reduced substantially. The parameters α and β are *compromise count factors* used in case of compromise match and its usage is elaborated in next section.

6. FEATURES OF PROPOSED MODEL

In previous section, we discuss how proposed model represents multifaceted constraints. In this section, we describe additional features supported by our proposed KR model.

Preferential Constraints: Our framework facilitates participants to indicate the relative importance among soft constraints, if any. For example, a participant can indicate $\text{facet}_1 > \text{facet}_5 > \text{facet}_3$ using an interface and appropriate priority values are assigned to the corresponding constraints. Changes in priority values ultimately affect similarity value.

Hidden Cost Constraints – We propose to penalize a profile with a hidden cost constraint, in the process of matchmaking. A hidden cost penalty is applied to the hidden cost constraint by reducing the priority value of the constraint to 0.9. This value is less than priority values of all remaining constraints (all other constraints have priority values of at least 1). Due to the penalty in terms of reduction in the priority, the similarity value of the profile that

contains a hidden cost constraint will be less than profiles that do not have a hidden cost constraint.

Symmetry/Non-symmetry: We introduce a parameter *omission penalty*, and its value can be set by a participant. This parameter value reduces resulting similarity value, for each constraint that is present in a Seller's profile but missing from a Buyer's profile; or vice versa.

If the value of an omission penalty is set to 0, a system shows characteristics of a symmetric matchmaking system, i.e. $Sim(P_i, P_j) = Sim(P_j, P_i)$. For any other value of an omission penalty such that $0 < \text{omission penalty} \leq 1$, a matchmaking system exhibits non-symmetric characteristics from buyers and sellers points of view.

Compromise match effect: As a compromise match is not an exact match, a similarity value between corresponding profiles should be reduced. In our matchmaking system, when there is a compromise match between two constraints, an intermediate similarity value (given by the function S in equation 1) is reduced by a certain factor. Consider an example of a soft constraint by a seller, "Prefer a non-smoker but ready to deal with a smoker" and a buyer's soft constraint as "I am looking an apartment where smoking is allowed but ready to rent a non-smoking apartment too". These two constraints have a compromise match. As both of the participants are ready to compromise with their preferred choices, it is likely that these two participants can reach an agreement. Hence a similarity value in case of a compromise match is influenced by the count (*compromise count*) of participants (one or both) willing to compromise.

We propose two *compromise count factors*, α and β to reduce a similarity value, in case of a compromise match. The values of α and β are set to less than 1. An intermediate similarity value is multiplied by these factors to obtain an expected reduction in a similarity value.

If a compromise count is one, then there are relatively less chances of an agreement as only one participant is ready to compromise. The factor α represents this case, while the factor β is used when compromise count is two.

We set the values of α and β such that a higher similarity value shall be resulted for a compromise match where both participants are ready to compromise and a lower similarity value shall be resulted if only one participant is ready to compromise.

7. EVALUATION

We have obtained results of the matchmaking system developed using our KR model for a house rental domain. Our system supports all the types of constraints discussed in section 2.1. The system generates an appropriate list of similarities among profiles. The system facilitates users to determine the ranking of matching profiles by tuning the values of parameters like the omission penalty and the compromise count factors.

8. CONCLUSION

Based on the study of various profiles, which are constituted as a result of multifaceted expectations of participants, we formally defined various constraint types that any matchmaking system should support. Our proposed new knowledge representation model represents complex constraints of participant profiles. We developed a matchmaking system based on our proposed KR

model for a house rental domain and obtained satisfactory results. We showed how our system supports all types of constraints which we defined as criteria for an effective matchmaking system. The effect of changes in the parameter values on the ranking of matching profiles is required to be analyzed in details.

9. ACKNOWLEDGEMENT

This work was partially supported by an ACEnet post-doctoral fellowship and an NSERC Grant of Virendra Bhavsar. His partial support from the International Institute of Information Technology, Pune, is also acknowledged.

10. REFERENCES

- [1] Baader, F., Calvanese, D., McGuinness, D. et al. 2003. The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, Cambridge, MA.
- [2] Bhavsar, V. C., Boley, H. and Yang, L. 2004. A Weighted-Tree Similarity Algorithm for Multi-Agent Systems in e-Business Environments. Computational Intelligence 20, 584-602.
- [3] Islam, M. R., Islam M. Z. and Nazia, L. 2008. A Tree-based Approach to Matchmaking Algorithms for Resource Discovery. International Journal of Network Management.
- [4] Kuokka, D. and Harada, L. 1996. Integrating Information via Matchmaking. Journal of Intelligent Information Systems 6, 261-279.
- [5] Joshi, Manish R., Bhavsar, V. C. and Boley, Harold. 2008. A Comparative Study of Features of Web-based Matchmaking Systems. Technical Report, Faculty of Computer Science, UNB, December 2008.
- [6] Li, L. and Horrocks, I. 2004. A Software Framework for Matchmaking Based on Semantic Web Technology. International Journal of Electronic Commerce 8, 39-60.
- [7] Mohaghegh, S. and Razzazi, M. R. 2004. An Ontology Driven Matchmaking Process. World Automation Congress 16, 248-253.
- [8] Noia, T. Di., Sciascio, E. Di., Donini, F. M. and Mongiello, M. 2004. A System for Principled Matchmaking in an Electronic Marketplace. International Journal of Electronic Commerce 8, 9-37.
- [9] Ragone, A., Straccia, U., Noia, T. Di. et al. 2007. Vague Knowledge Bases for Matchmaking in P2P E-Marketplaces. In proceedings of the ESWC.
- [10] Bassiliades, Nick., Antoniou, G., Vlahavas, I. 2004. A Defeasible Logic Reasoner for the Semantic Web, In Rules and Rule Markup Languages for the Semantic Web 3323/2004, 49-64.
- [11] Subrahmanian, V. S., Bonatti, P., Dix, J. et al. 2000. Heterogeneous Agent Systems. MIT Press.
- [12] Sycara, K., Widoff, S., Klusch, M. and Lu, J. 2002. Larks: Dynamic Matchmaking among Heterogeneous Software Agents in Cyberspace. Autonomous Agents and Multi-Agent Systems 5, 173-203.
- [13] Veit, D., Müller J. P. and Weinhardt, C. 2002. Multidimensional Matchmaking for Electronic Markets. International Journal of Applied Artificial Intelligence 16, 853-869.